

## Association for Information Systems AIS Electronic Library (AISeL)

---

AMCIS 1998 Proceedings

Americas Conference on Information Systems  
(AMCIS)

---

December 1998

# Mapping Maturity Characteristics into PSP

Anita La Salle  
*American University*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1998>

---

### Recommended Citation

La Salle, Anita, "Mapping Maturity Characteristics into PSP" (1998). *AMCIS 1998 Proceedings*. 295.  
<http://aisel.aisnet.org/amcis1998/295>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Mapping Maturity Characteristics into PSP

Anita J. La Salle

Department of Computer Science and Information Systems  
American University

## Abstract

*The Personal Software Process (PSP)<sup>sm</sup> is a technique designed to enable software engineers to improve their productivity and the quality of their work. PSP focuses on the habits of an individual producer and the ability of the individual to capture data that will help improve future performance. The Capability Maturity Model (CMM)<sup>sm</sup> is a framework that focuses on organizations and the processes that organizations can put in place to improve their ability to sustain high quality software production. This paper describes a mapping between CMM and PSP techniques across organizational activities with particular attention to the contribution that PSP might make to an organization's maturity level.*

*Note: The Personal Software Process, PSP, and The Capability Maturity Model for Software, CMM, developed by the Software Engineering Institute at Carnegie Mellon University, are service marks of Carnegie Mellon University.*

## Background: What is PSP?

PSP prescribes a sequence of activities that aid software engineers in developing a disciplined approach toward producing systems based upon measurements of personal performance over time.

The principles of PSP are not really new. The PSP approach has foundations in practices used in project management and statistical process control and has traditionally been a component of good engineering habits.

What does PSP involve? PSP is a formal methodology that focuses on the individual software engineer's work habits and involves the individual in processes that will lead to improved product quality and personal productivity. The logical extension of the effect of PSP is that team and organizational performance will be enhanced by individual performance enhancements.

When applying PSP, software engineers use standardized forms and techniques to:

- Track how they spend their time to gain insight into how to manage their time.
- Develop benchmarks for projecting time estimates for future projects and techniques for tracking time spent on project tasks.
- Gather data about their personal productivity with respect to program size and how to use that data to project future development times and to serve as a basis for tracking their work.
- Gather data on product defects that can be used to prevent, find, and repair defects in future projects.
- Develop code-review practices and gather review data, using check-sheets, to improve defect detection, improve product quality by reducing the number of defects injected into a system, and improve defect removal rates.
- Analyze defect types to reduce the number of design defects
- Understand how to scale-up data obtained from small projects to be able to project quality and productivity into large-scale projects.

PSP focuses the individual software engineer's attention on self-improvement through the application of well-defined methodologies.

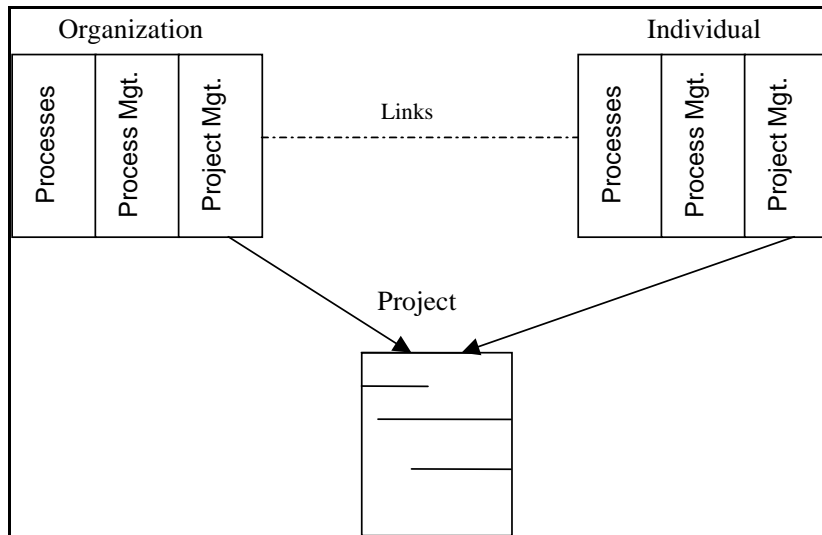
## Background: What is CMM?

CMM is a process-improvement framework that focuses on the organization. CMM describes key elements of effective software processes that, when followed, may lead an organization from an undisciplined, ad hoc state of software production to a "mature" disciplined state.

CMM defines five levels of organizational maturity:

- Initial (Level 1) - few processes are formalized or defined, software development is chaotic, success relies on heroic efforts.
- Repeatable (Level 2) - processes exist for managing software projects and performance in subsequent projects is predictable based upon recorded experience with similar prior projects.
- Defined (Level 3) - the organization has an integrated set of software processes that are documented and standardized and those processes are applied across the organization. "The Software Process Capability of Level 3 organizations can be summarized as standard and consistent because both software engineering and management activities are stable and repeatable." [2]
- Managed (Level 4) - the organization maintains a software process database to record, manage, and analyze quantitative data about the organization's processes and projects. Because processes and projects are measured, the organization's performance is quantifiable and predictable.

- Optimizing (Level 5) - "At the Optimizing Level, the entire organization is focused on continuous process improvement."  
[2] All aspects of software quality are tracked and processes are refined to reduce software defects and rework.  
CMM focuses the organization's attention and practices on those activities across the organization likely to result in high quality software.



**Figure 1. Integrating Organizational and Personal Processes**

## Relationship Between PSP and CMM

PSP and CMM are related in the sense that both are quality-focused and CMM creates an organizational environment where an individual's application of PSP is consonant with organizational goals.

Figure 1 graphically depicts how the two approaches impact on software projects. Mature organizations (i.e., those that subscribe to the tenets of CMM) have defined processes; procedures are in place to manage the processes; and, the application of those processes to a specific software project results in a track-able project-management scheme for the specific project underway. An individual software engineer (particularly one who subscribes to PSP habits) will manage personal processes and apply them to the generation of software for a specific project. Both PSP and

CMM are provenly effective. Where the problem arises is that while CMM and PSP both contribute to an organization's potential for producing quality software, there are few specific links between the two approaches.

## Linking CMM and PSP

Activities prescribed by PSP fit into practices at several CMM maturity levels but most predominantly at CMM Level 2 where basic software management controls reside.

What is needed is a way to encapsulate PSP-like activities in every CMM maturity level so that all project practitioners gain detailed insight into the processes applied to each project.

For each CMM maturity level, there are "key process areas" that organizations view as foci of their activities to improve software quality. For example, in Level-2 one key process area refers to how system requirements are managed:

Requirements Management activities exist in the organization for documenting, reviewing, managing, tracking, and altering user requirements before, during, and after product development.

Associated with each key process area are "key practices" that describe what should be done within an organization to promote the institutionalization of the key process area. CMM does not prescribe how key practices are implemented. It concentrates on principles of implementation and leaves each organization to translate the key practices into their own systems development framework. So, for Level-2's Requirements Management Key Process Area, CMM only specifies that:

Requirements are documented and reviewed, responsibility is established for analyzing the requirements and determining how they will be allocated to hardware and software, competent individuals manage the requirements and are provided with the tools to accomplish this, the requirements become the basis for the software development plans, change of requirements is managed, and the implementation is verified.

CMM's template for presenting key processes and practices permits flexible interpretation, but at the same time, their adoption by an organization accomplishes several critical objectives: operational policies and procedures are established along with leadership roles that together will ensure that the processes will endure; organizational structures are established and resources are allocated so that processes can be accomplished; and plans are put in place, procedures, requirements and products are documented, and oversight is carried out to insure that the key process area is actually implemented.

Even though CMM concentrates on organizations, managers and teams, individual PSP practices naturally lend themselves to key process areas. For example, the formal, disciplined nature of PSP (e.g., maintaining personal records of activities, reviewing and distilling those records for trends and personal improvement) provides a framework for an individual's role in a team responsible for establishing, documenting, analyzing, and allocating system requirements in Level-2. In particular, the disciplined personal work-style of a team member is critical to the "repeatability" of this level of maturity. In addition, the role of the manager at this level can expand so individuals are explicitly enabled to apply the PSP approach to each key process area.

Table 1 depicts how the CMM and PSP can be structurally integrated to link the advantages of each with the other for all CMM maturity levels. In the left-most columns are the CMM maturity levels with their key processes and the key practices that

enable those maturity levels. These are individually mapped to personal practices or activities managed by individual team members. For every key organizational practice, there is a compatible personal activity that complements the organization's practice. This approach differs considerably from PSP's current application. As PSP is currently defined, activities are confined to the production of software and this is narrowed to a time-management, defect management, productivity perspective. Mapping CMM to PSP expands the scope of the software engineer's processes.

**Table 1. Integrating CMM and PSP**

Organizational Focus (CMM)	Maturity Focus, Key Processes, and Key Practices			Individual Focus (PSP)	
				Time Management (Productivity)	Product Management (Quality)
	Level-2	Requirements Management	Key Practice-1 Key Practice-2 ...	PSP activities	PSP activities
		Software Project Planning	Key Practice-1 Key Practice-2 ...	PSP activities	PSP activities
		...	...		
	Level-3	Organizational Process Focus			
		Organizational Process Definition			
		...			
	Level-4				

### Conclusions

According to Paulk [2], "The CMM ...does not address all of the issues that are important for successful projects. ... CMM does not ... suggest how to select, hire, motivate, and retain competent people. ... The CMM represents a "common sense engineering" approach to software process improvement." Likewise, PSP proposes an engineering approach for individuals to apply to themselves to improve their competence. This paper advocates that integrating the two strengthens the effectiveness of both CMM and PSP.

### References

1. Humphrey, Watts S., "Introduction to the Personal Software Process." Addison-Wesley, 1997.
2. Paulk, Mark c., et. Al., "The Capability Maturity Model: Guidelines for Improving the Software Process." Addison-Wesley, 1995.

Note: Additional references available upon request.